LEVEL III (12) AD-E000 471

NRL Memorandum Report 4172

ADA087211

A Modified Prony Method Approach to Echo-Reduction Measurements of Time-Limited Transient Signals

DAVID H. TRIVETT

Measurements Branch
Underwater Sound Reference Detachment
P. O. Box 8337 *Orlando, Florida* 32856

June 16, 1980

NAVAL RESEARCH LABORATORY
Washington, D.C.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 4172	2. GOVT ACCESSION NO. AD-A087211	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A MODIFIED PRONY METHOD APPROACH TO ECHO-REDUCTION MEASUREMENTS OF TIME-LIMITED TRANSIENT SIGNALS		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David H. Trivett		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Underwater Sound Reference Detachment P.O. Box 8337, Orlando, Florida 32856		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 63562N; 82-0595-0-0
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Research Laboratory Underwater Sound Reference Detachment P.O. Box 8337, Orlando, Florida 32856		12. REPORT DATE June 16, 1980
		13. NUMBER OF PAGES 65
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Prony method
Echo reduction
Complex exponential expansion
Acoustic panel measure
Transient signals

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A modified Prony method is presented for measuring the steady-state echo reduction of acoustic panels. The method extrapolates the steady-state amplitudes from the transient portion of the signal allowing time-limited measurements. Th method is applied to measurements of square panels 76 cm on an edge and 0.95 cm-thick steel and aluminum in the frequency range of 3 - 10 kHz. The signals were time-limited to 200 μs (0.6 - 2.0 λ) by the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE UNCLASSIFIED (Abstract continues)
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Abstract continued)

arrival of the diffracted signal from the panel edges. Results are compared with theoretical values and a program listing is included.

CONTENTS

	PAGE
INTRODUCTION	1
THEORY	2
COMPARISON OF MODIFICATIONS	9
SELECTION OF INPUT VARIABLES	11
Data Increment	11
Order of Expansion	14
Time Window	15
EXPERIMENTAL PROCEDURE	16
EXPERIMENTAL RESULTS	20
CONCLUSION	24
ACKNOWLEDGMENTS	24
REFERENCES	24
APPENDICES	
A - Overview	26
B - List of Variables in Prony	27
C - Input Example	29
D - Output Example	31
E - Prony Listing	33

A MODIFIED PRONY METHOD APPROACH TO ECHO-REDUCTION MEASUREMENTS OF TIME-LIMITED TRANSIENT SIGNALS

INTRODUCTION

The current procedure for making echo-reduction measurements of acoustic panels at the Underwater Sound Reference Detachment of the Naval Research Laboratory employs steady-state signal processing of measurements made in the anechoic tank. The anechoic tank allows measurements to be performed at various temperatures and hydrostatic pressures. However, it restricts panel width to 76.2 cm due to the size of the access port.

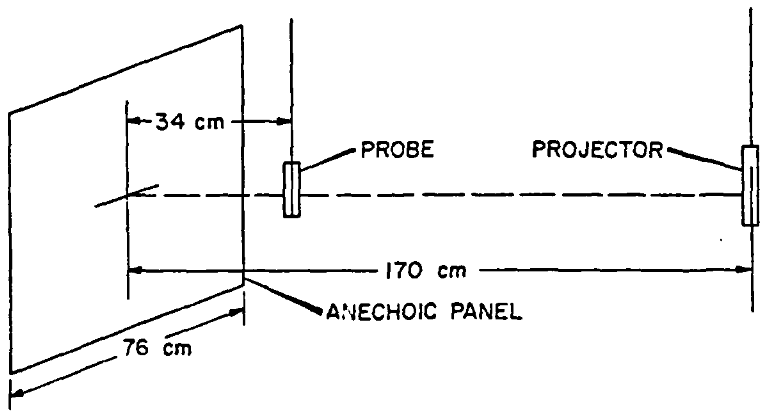

Fig. 1 - Panel measurement geometry

Figure 1 illustrates the geometry of the measurements. The panel is insonified by a projector 170 cm in front of the panel, and the incident and reflected signals are monitored by a probe 34 cm in front of the panel. For a step sinusoid signal from the projector, the output of the probe, illustrated in Fig. 2, consists of 426-μs segment of incident signal followed by a 141-μs segment of both incident and reflected signals followed by the arrival of the diffracted signal from the panel edges. In order to eliminate the undesired diffracted signal, the measurements are time limited to approximately 140-μs for the reflected signal. This time limit and the constraint that the signal reach its steady-state response currently restrict measurements to frequencies above 15 kHz. To obtain measurements below 15 kHz it must be possible to extrapolate the steady-state response from the transient portion of the signal.

Manuscript submitted March 30, 1980.

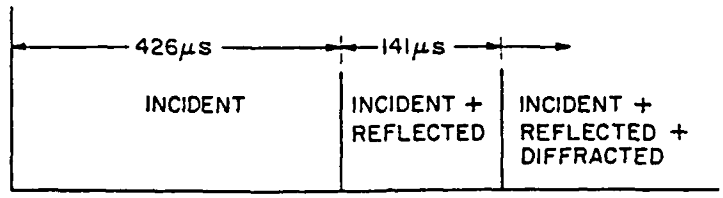

Fig. 2 - Time segment output of probe for step sinusoid signal

The Prony [1] method first published in 1795 allows such an extrapolation and has been investigated as a means of allowing low-frequency panel measurements. In this report a modified Prony method is presented, which allows measurements to be performed down to 2 kHz, along with the results of measurements on various panels to illustrate the ability of the modified approach.

THEORY

Any signal that can be represented by a function that is the solution of a set of constant-coefficient linear differential equations has an expansion given by [2]

$$f(t) = \sum_{j=1}^{N} A_j \exp(s_j t) , \qquad (1)$$

where N is the order of expansion and A_j and s_j are respectively the amplitudes and poles to be determined. Since $f(t)$ is real, both A_j and s_j must be real or appear in complex conjugate pairs. The real part of the pole s_j is the time constant associated with the transient response, and the imaginary part is the angular frequency of the jth component. The phase information is contained in the amplitudes A_j.

There are two immediate difficulties in obtaining the expansion in Eq. (1). First there is the problem of the nonlinearity of the expansion in s_j and second, there is the indeterminacy of the order of expansion N. The first problem was solved by Prony and his method will be described below. The indeterminacy of the order of expansion will be discussed after the basic equations have been derived.

Prony found that the desired expansion could be obtained by considering a uniformly sampled version of $f(t)$. Equation (1) then becomes

$$f(k\Delta) = \sum_{j=1}^{N} A_j \exp(s_j k\Delta) , \qquad (2)$$

where Δ is the data sampling time interval and k is an indexing integer running from 0 to M such that $M\Delta$ is equal to the observation time.

Let $$z_j = exp(s_j \Delta) \cdot \tag{3}$$

Then $$f(k\Delta) = \sum_{j=1}^{N} A_j z_j^{k} . \tag{4}$$

Equation (4) does not represent a linear set of equations in z_j. However, Prony noted that solutions to z_j could be found by first defining an Nth order polynomial in z whose roots are z_j. That is

$$P(z) = \sum_{i=0}^{N} a_i z^{i} = \prod_{j=1}^{N} (z-z_j), \tag{5}$$

where $\alpha_N = 1$. The α_i coefficients have one more degree of freedom than the z_j; hence the arbitrary setting of α_N equal to unity. Using Eqs. (4) and (5) to evaluate the following expression:

$$\sum_{i=0}^{N} a_i \, f[(i + \ell)\Delta]. \tag{6}$$

For $\ell = 0$ to M-N yields

$$\begin{aligned}\sum_{i=0}^{N} \alpha_i \, f[(i + \ell)\Delta] &= \sum_{i=0}^{N} \alpha_i \left(\sum_{j=1}^{N} A_j z_j^{\,i+\ell} \right) \\ &= \sum_{j=1}^{N} A_j z_j^{\ell} \left(\sum_{i=0}^{N} \alpha_i z_j^{i} \right) \\ &= \sum_{j=1}^{N} A_j z_j^{\ell} \, P(z_j) = 0.\end{aligned}$$

Or $$\sum_{i=0}^{N-1} \alpha_i \, f[(i + \ell)\Delta] = -f[(N + \ell)\Delta] \tag{7}$$

since $\alpha_N = 1$. Equation (7) yields M-N+1 linear equations in the N unknown polynomial coefficients. For $M=2N-1$ the equations can be solved to obtain the α_i which completely define the polynomial in Eq. (5). The roots of the polynomial can be found to provide the required z_j. These z_j are then substituted into Eq. (4) to determine the amplitudes A_j. The expansion is then completed by transforming the z_j back into the complex s plane using

$$s_j = \Delta^{-1} \ln z_j . \tag{8}$$

Prony's method isolates the nonlinearity of the expansion to Eq. (5). However, the method introduces a third difficulty through Eq. (8).

Negative real z_j cannot be transformed back into the complex s plane. Although these represent nonphysical terms, noise contaminated data will occasionally produce negative real z_j. When this occurs, one must be satisfied with the expansion in the form of Eq. (4).

In addition to negative real z_j, nonphysical roots occur as a consequence of the indeterminacy of the order of expansion. If fewer poles are specified in the expansion than actually exist in the data, then the poles returned by the algorithm deviate from the true poles and, in most cases, the true poles will not be returned at all. If more poles are specified than actually exist, the algorithm will return extraneous poles in addition to a set that deviates from the true poles. The presence of the extraneous poles adversely affects the calculated amplitudes of the true poles. Attempts to systematically determine the order of expansion by checking for linear dependent columns in the matrix generated by the Prony difference equations or by analyzing the eigenvalues of the matrix have failed in the presence of noise [3].

In the modified method to be developed in the remainder of this section, the problems of negative real z_j and the indeterminacy of the order of expansion will be circumvented. This will be accomplished through the introduction of a priori poles and will be discussed after the equations are developed. However, first a modified least square extension of the Prony method will be developed.

The solution of Eq. (7) requires at least $2N$ sampled data points. For the case where $M = 2N-1$, the solution will pass directly through the sampled data points since the problem is exactly specified. However, in general, the number of available data points exceeds $2N$ and a type of least square solution is used. This is most conveniently done by performing a pseudo-inverse of Eqs. (7) and (4). Equation (7) in matrix notation is

$$Ax = y, \tag{9}$$

where

$$A = \begin{pmatrix} \delta_0 & \delta_1 & \delta_2 & \cdot\ \cdot\ \cdot\ \cdot & \delta_{N-1} \\ \delta_1 & \delta_2 & \delta_3 & \cdot\ \cdot\ \cdot\ \cdot & \delta_N \\ \delta_2 & \delta_3 & \delta_4 & \cdot\ \cdot\ \cdot\ \cdot & \delta_{N+1} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \delta_{M-N} & \delta_{M-N+1} & \delta_{M-N+2} & & \delta_{M-1} \end{pmatrix},$$

$$x = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_{N-1} \end{bmatrix},$$

and

$$y = - \begin{bmatrix} \delta_N \\ \delta_{N+1} \\ \delta_{N+2} \\ \cdot \\ \cdot \\ \cdot \\ \delta_M \end{bmatrix}.$$

The pseudo-inverse is formed by premultiplying both sides of Eq. (9) by the transpose of A. That is,

$$A^T A x = A^T y. \tag{10}$$

The matrix $A^T A$ is a symmetric $N \times N$ matrix. Writing out Eq. (10) explicitly yields

$$\sum_{i=0}^{N-1} a_i \left\{ \sum_{k=0}^{M} \delta[(i+k)\Delta] \; \delta[(\ell+k)\Delta] \right\}$$

$$= - \sum_{k=0}^{M} \delta[(N+k)\Delta] \; \delta[(\ell+k)\Delta] \tag{11}$$

for $\ell = 0$ to $N-1$.

Following the same procedure the least square extension of Eq. (4) is

$$\sum_{j=1}^{N} A_j \left\{ \sum_{k=0}^{M} (z_i z_j)^k \right\} = \sum_{k=0}^{M} z_i^k \delta(k\Delta) \tag{12}$$

for i = 1 to N.

This least square extension of the Prony method was examined by McDonough [4], who observed that the normal equations become increasingly ill conditioned as the sampling rate increases. The source of the difficulty lies in the data interval spanned by each difference equation. The interval spanned is $N\Delta$; and as the sampling rate $1/\Delta$ increases, the spanned interval decreases. As smaller portions of the signal are used in each difference equation, the noise in the signal has a greater deleterious effect on the pole estimates. In order to eliminate this effect Beatty [5] uses every pth sampled data point instead of adjacent samples to satisfy the difference equations. This results in a data span of $Np\Delta$ for each difference equation. Then, as the sampling rate is increased, the value of p is increased to maximize the interval spanned. All of the data are used since each difference equation's initial data sample is $\delta(k\Delta)$ where k is an incrementally stepped integer. For this modification the equations are developed below.

Let

$$\delta(k\Delta + np\Delta) = \sum_{j=1}^{N} A_j \exp[s_j(k\Delta + np\Delta)], \tag{13}$$

where k, n, and p are all positive integers and p is the introduced data increment. Let

$$z'_j = \exp(s_j p\Delta) \tag{14}$$

and

$$\delta(k,n) = \delta(k\Delta + np\Delta). \tag{15}$$

Then Eq. (13) becomes

$$\delta(k,n) = \sum_{j=1}^{N} A_j \exp(s_j k\Delta)\, z'^{\,n}_j = \sum_{j=1}^{N} B_j z'^{\,n}_j , \tag{16}$$

where $B_j = A_j \exp(s_j k\Delta)$. Since the difference equations do not depend upon B_j (k is a constant in each difference equation), the difference equations are

$$\sum_{n=0}^{N-1} \delta(k,n) a_n = -\delta(k,N) \tag{17}$$

for $k = 0$ to $M-N$. The least square extension of Eq. (17) is obtained by forming the pseudo-inverse resulting in

$$\sum_{n=0}^{N-1} \alpha_n \left\{ \sum_{k=0}^{M-N} \delta(k,n)\delta(k,\ell) \right\} = -\sum_{k=0}^{M-N} \delta(k,N)\delta(k,\ell) \tag{18}$$

for $\ell = 0$ to $N-1$. The solution of Eq. (18) yields the coefficients α_n which define $P(z')$. The roots of $P(z')$ are z'_j. The z'_j may be transformed back into the complex s plane using

$$s_j = (p\Delta)^{-1} \ln z'_j. \tag{19}$$

Once the s_j are computed they may be substituted into Eq. (3), and then Eq. (12) can be used to obtain the amplitudes A_j. As before, the procedure fails if a negative real z'_j is obtained; except in this case not even the amplitudes can be obtained.

In the case of echo-reduction measurements the useful information is contained in the amplitude of the steady-state driving frequency component. Under this circumstance the Prony method can be modified by the introduction of a priori poles to circumvent the problems of negative real z'_j and the indeterminacy of the order of expansion. If the algorithm is constrained to find the correct a priori poles, the remaining poles used in the expansion are used as curve-fitting poles. There is no requirement that the curve-fitting poles have any physical significance. Then the only requirement on the order of expansion is that there be a sufficient number of curve-fitting poles such that the mean square deviation between the waveform and it's expansion be below some arbitrary small value.

The problem of negative real z'_j is eliminated by using an odd integer for the data increment p. Then Eq. (12) may be used directly to obtain the amplitudes since

$$z_j = (z'_j)^{1/p}, \tag{20}$$

and negative real z'_j are handled with

$$z_j = -|z'_j|^{1/p}. \tag{21}$$

The modified equations are developed below.

Let

$$\sum_{\ell=0}^{R} b_\ell z'^{-\ell} = \prod_{j=1}^{R} (z' - z'_j) = N(z') \tag{22}$$

be the polynomial generated by the R a priori poles where $b_R = 1$. Since $N(z')$ must be a factor of Prony's polynomial, $P(z')$ in Eq. (5) can be written as

$$P(z') = N(z') \sum_{n=0}^{N-R} \beta_n z'^{-n} ,$$

where $\beta_{N-R} = 1$. Writing Eq. (23) explicitly yields

$$\sum_{i=0}^{N} \alpha_i z'^{-i} = \sum_{\ell=0}^{R} b_\ell z'^{-\ell} \cdot \sum_{n=0}^{N-R} \beta_n z'^{-n} .$$

Equating similar powers of z' yields

$$\alpha_i = \sum_{\ell=0}^{R} b_\ell \, \beta_{i-\ell} , \tag{25}$$

where $\beta_N = \beta_{N-1} = \cdots = \beta_{N-R+1} = 0$ and $\beta_{-1} = \beta_{-2} = \cdots = \beta_{-R} = 0$. Substitution of Eq. (25) into Eq. (17) yields

$$\sum_{i=0}^{N-1} \delta(k,i) \left\{ \sum_{\ell=0}^{R} b_\ell \beta_{i-\ell} \right\} = -\delta(k,N), \tag{26}$$

which may be rewitten as

$$\sum_{i=0}^{N-R-1} \beta_i \left\{ \sum_{\ell=0}^{R} b_\ell \delta(k,i+\ell) \right\} = - \sum_{j=0}^{R} b_j \delta(k,N-R+j). \tag{27}$$

Let

$$F(k,n) = \sum_{j=0}^{R} b_j \delta(k,n+j). \tag{28}$$

Then Eq. (27) becomes

$$\sum_{i=0}^{N-R-1} \beta_i \, F(k,i) = -F(k,N-R). \tag{29}$$

Forming the pseudo-inverse results in

$$\sum_{i=0}^{N-R-1} \beta_i \left\{ \sum_{k=0}^{M-N} F(k,i) F(k,j) \right\} = - \sum_{k=0}^{M-N} F(k,N-R) F(k,j) \tag{30}$$

for j = 1 to N-R-1. Equation (30) is solved to obtain the coefficients B_i. These B_i are then used to generate the reduced Prony polynomial whose roots z_j' are the curve-fitting poles. The curve-fitting poles together with the a priori poles are transformed using Eqs. (20) and (21). The computed z_j are then used in Eq. (12) to find the amplitudes A_j. The amplitude of the steady-state driving frequency pole is then used to compute the echo reduction in a manner to be discussed later.

COMPARISON OF MODIFICATIONS

In order to test the effectiveness of the various modifications on the type of signal to be encountered in panel measurements, the waveform in Fig. 3 was generated. The waveform simulates the reflection of a 3-kHz step sinusoid from a 0.95-cm thick infinite steel plate. It was computer generated by successively adding, with suitable time delays, the multiple internal reflections that are transmitted back through the face of the plate. Seven data files were constructed by sampling the waveform at 1 MHz and adding various levels of random noise. The first 200 μs of each data file was then analyzed by the Prony method in six different manners.

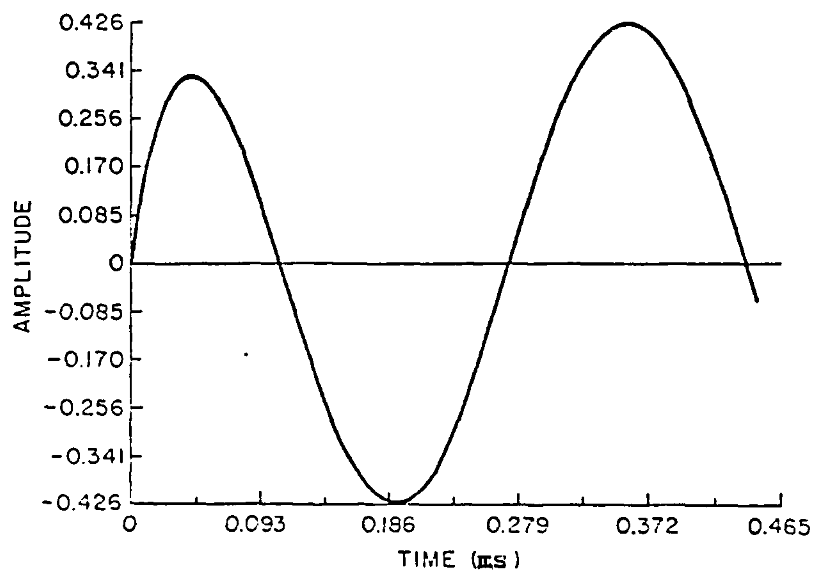

Fig. 3 - Simulated reflection of a 3-kHz step sinusoid from a 0.95-cm thick infinite steel plate

The six expansions performed were divided into two sets of three, one of which used three poles in the expansion while the other used fifteen poles. Three pole expansions were used since the waveform has a known three-pole expansion consisting of a complex conjugate pair representing the steady-state driving frequency and a real pole associated with the transient response of the plate. The choice of fifteen poles was arbitrary. Each set contained one expansion with no a priori poles and a data increment of one one expansion with no a priori poles and a data increment of eleven, and an expansion with two a priori poles and a data increment of eleven. The a priori poles entered were the steady-state driving frequency poles and since 200 data points were used, all expansions used least square methods.

The amplitude of the steady-state driving frequency poles or the poles closest to the driving frequency when no a priori information was entered were used as a measure of the accuracy of the expansion. The results are plotted in Fig. 4 where the correct amplitude is 0.426 and the two expansions with a data increment of one were not plotted since the expansions failed in most cases to obtain any poles close to the driving frequency. The results indicate that unless one has a high signal to noise ratio the only method that obtains useful information is the use of both a priori and curve fitting poles. In general the more a priori information supplied and the more curve fitting poles used the better the results.

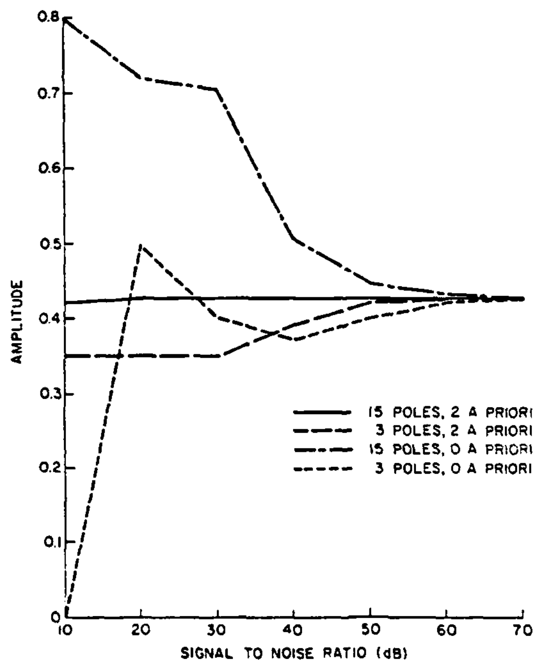

Fig. 4 - Amplitudes of 3-kHz component of simulated panel reflection obtained by Prony algorithm with various modifications and noise levels. Correct amplitude is 0.426.

SELECTION OF INPUT VARIABLES

The modified Prony algorithm contains three user supplied variables: the data increment, order of expansion, and length of time window. Unfortunately, no strict limits on the variables can be set since they depend on the complexity of the waveform being analyzed and the signal-to-noise ratio. However, given the constraints under which the method will be used in making echo-reduction measurements, several useful comments can be made. In this section the required values of the variables for echo-reduction measurements will be investigated.

Data Increment

In order to investigate the variables with the actual signals to be encountered in panel measurements, a set of waveforms was obtained for the reflection of a step sinusoid from a 1.27-cm-thick, 76-cm-square steel panel. The waveforms were obtained by placing a probe 5 cm and a projector 170 cm in front of the steel panel. A 1-ms pulse from the projector produced a 67-μs segment of direct signal, at the probe, followed by 250 μs of incident plus reflected signals before the arrival of the diffracted signal from the panel edges. The output voltage of the probe was sampled at 1 MHz, and 100 separate recordings were averaged to reduce the incoherent noise level. A second set of measurements were made without the steel panel in place to obtain a long recording of the incident signal. The two waveforms were then directly subtracted to yield the reflected waveform from the steel plate. Figure (5) illustrates the waveform of the steel panel reflection at 3 kHz.

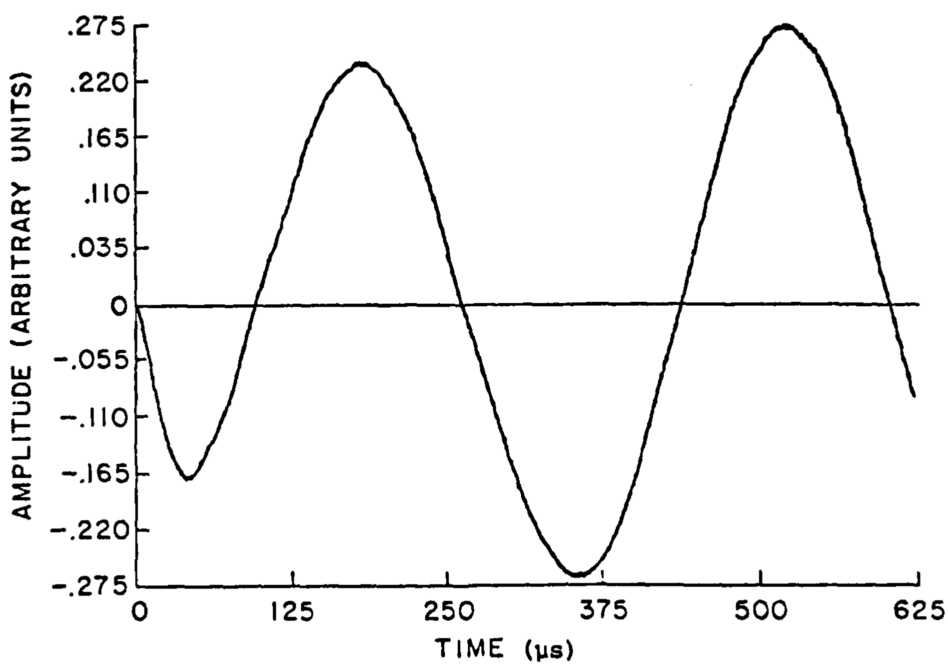

Fig. 5 - Waveform obtained by the reflection of a 3-kHz step sinusoid from a 1.27-cm-thick steel plate

The modified Prony method was used to analyze the 3-kHz waveform to obtain the amplitude of the steady-state driving frequency poles that were entered a priori. A 15-pole expansion was used, and the time window was reduced in 10-μs steps from 250 to 110 μs. Four different data increments were used, and the results are illustrated in Fig. (6). Values for the three largest data increments do not span the entire time scale. This is due to the requirement of a minimum number of data points for the expansion as determined by the order of expansion and the data increment. The minimum number of data points required by the algorithm is given by

$$\text{Min. \# pts.} = (\text{Data Increment} + 1) * \text{Order of Expansion.} \tag{31}$$

With a 15-pole expansion the minimum number of data points for a data increment of 11 is 180. Since the waveform was sampled at 1 MHz, the minimum time window for a data increment of 11 is 180 μs.

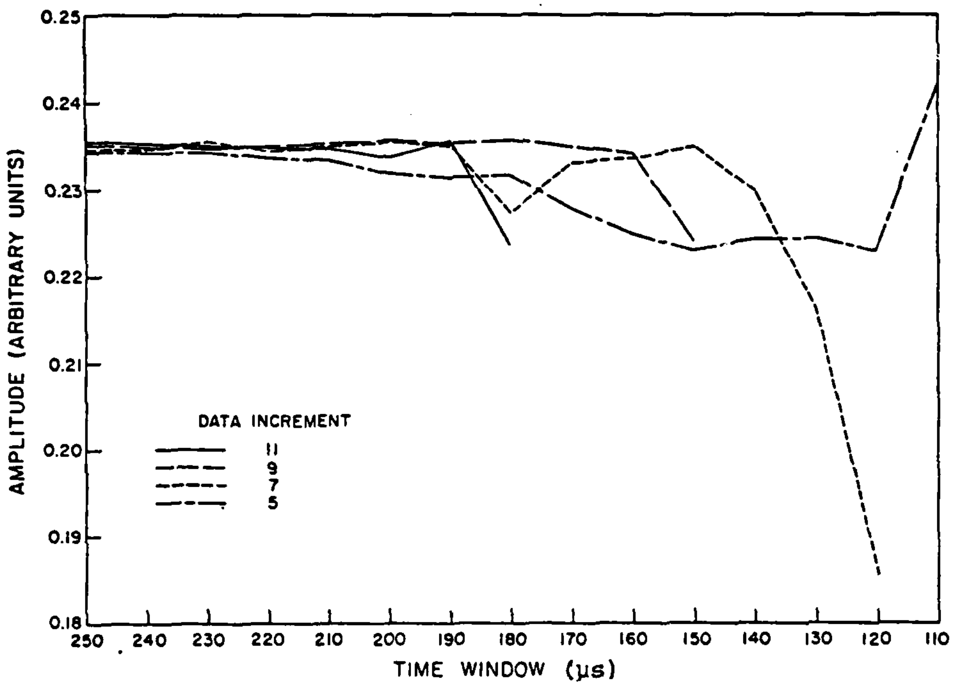

Fig. 6 - Amplitude of the steady-state driving frequency poles for waveform in Fig. (5) obtained with 15-pole expansions and various data increments

The results in Fig. (6) are consistent for large time windows but vary as the time window is reduced. Since in general the largest possible data increment should be used, the appropriate portions of Fig. (6) have been reproduced in Fig. (7).

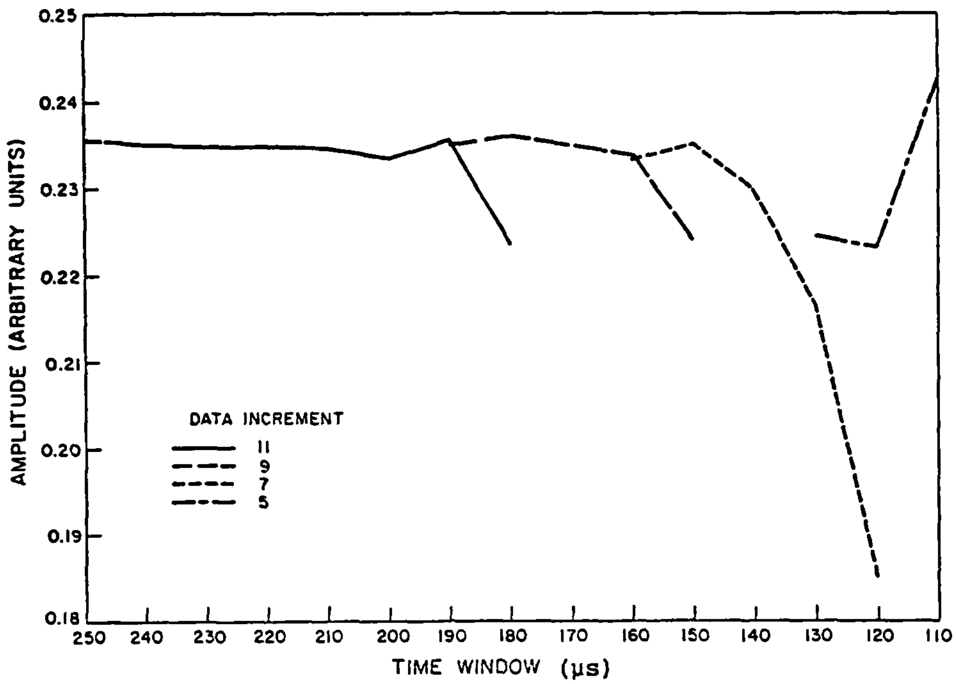

Fig. (7) - Segments of data in Fig. (6) illustrating effect of minimum time window

Figure (7) illustrates the one exception to the largest possible data increment rule. The values at 180 μs for a data increment of 11 and at 150 μs for a data increment of 9 show marked deviations from the average value. Both of these cases correspond to minimum values for the time window of the associated data increment. The difficulty lies in the matrix generated from the Prony difference equations. If no a priori poles are used, the minimum number of points in Eq. (31) generates an $n \times n$ matrix where n is the order of expansion. Since this is a square matrix, no least-square technique is required to solve the equations. However, if a priori poles are used, the matrix generated is an $(n-r) \times n$ matrix where r is the number of a priori poles used in the expansion. The matrix represents an overdetermined set of equations for $n-r$ unknown coefficients, and a least-square technique must be used to solve the equations. When r is small in comparison to n or when the matrix is nearly a square matrix, the least-square technique introduces considerable error into the calculation. All of the measurements in Figs. (6) and (7) were done with an order of expansion of 15 and with two a priori poles. By reducing the value of the data increment when the minimum time window is approached, the matrix is no longer nearly a square matrix, and the least-square technique returns consistent values as illustrated in Fig. (7). Thus the largest data increment should be used except when the minimum time window is approached, and then the next lower value should be used.

Figure (7) also indicates that for time windows of less than 150 μs an insufficient portion of the waveform is being used to yield useful results.

Order of Expansion

The order of expansion is the most difficult variable to determine. There must be a least as many poles in the expansion as there are in the signal. However, a knowledge of the structure of the signal does not guarantee correct results. As illustrated in Fig. (4) a 3-pole expansion with two poles entered a priori was sufficient for a waveform that was known to have only three poles when the signal-to-noise ratio was 50 dB. When the signal-to-noise ratio was lowered, the 3-pole expansion yielded incorrect results.

In general the more poles used in the expansion together with the a priori poles, the better the results. Unfortunately the larger the order of expansion, the longer the running time for the program. The optimum value is strongly dependent on the complexity of the waveform, the signal-to-noise ratio, and the length of the time window. To get some idea of the required order of expansion for echo-reduction measurements, the waveform in Fig. (5) was again analyzed. Five different orders of expansion were used on time windows that varied from 250 to 150 µs. The data increment was determined by the results of the previous section.

In Fig. (8) the amplitude of the steady-state driving frequency pole, which was entered a priori, has been plotted against the length of the time window. The 15-pole expansion deviates by less than 0.1 dB over the entire time window span. The 12-pole expansion is consistent down to a time window of 180 µs. However, the remaining expansions yield inconsistent results and vary from one time window to the next. This indicates that the order of expansion must be at least 15 for echo-reduction measurements of simple homogeneous plates and may have to be higher for nonhomogeneous plates.

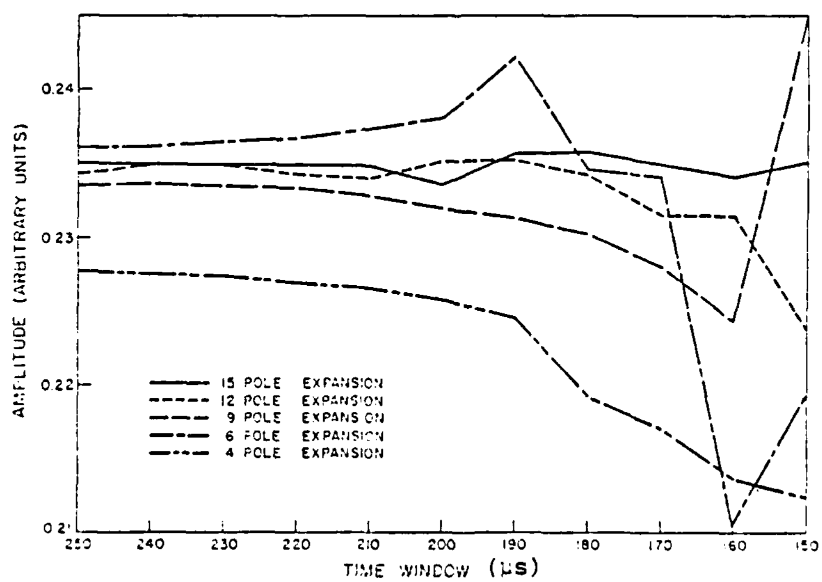

Fig. (8) - Amplitude of the steady-state driving frequency poles for waveform in Fig. (5) obtained with various orders of expansion

Time Window

The minimum time window necessary for the algorithm to yield useful results is dependent on the acceptable error and the signal-to-noise ratio. Since the signal-to-noise ratio is a function of the reflection coefficient of the panel the error associated with a particular time window will vary from one panel to another. In order to obtain some idea of the minimum time window, the steel panel described earlier was investigated.

Three waveforms were obtained for the reflection of a step sinusoid from the steel panel at frequencies of 2, 2.5, and 3 kHz. The signal-to-noise ratio in each was approximately 40 dB. Each waveform was analyzed with a 15-pole expansion that included two steady-state driving frequency poles entered a priori. The time window was varied from 250 to 130 μs in 10-μs steps.

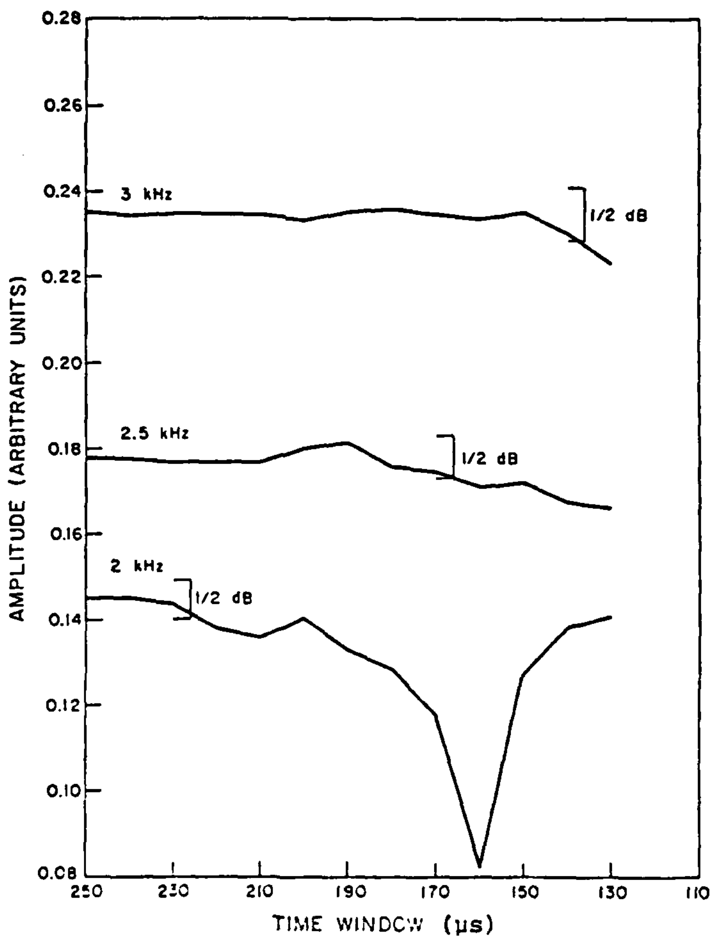

Fig. (9) - Amplitude of the steady-state driving frequency poles as a function of time window

In Fig. (9) the amplitude of the driving frequency pole has been plotted against the length of the time window for each of the waveforms. Since the actual amplitudes are unknown, the average values of the flat portions of the curves were used as the correct amplitudes. Arbitrarily choosing an allowable error of 0.25 dB from the average value as a measure of the accuracy of the algorithm produced minimum time windows of 140 μs for the 3-kHz waveform, 170 μs for the 2.5-kHz waveform, and 230 μs for the 2-kHz waveform. These time windows correspond to 0.42, 0.425, and 0.46 wavelengths, respectively, for 2.5 and 2 kHz. Since these values are in good agreement, a general rule of approximately half a wavelength as the minimum time window has been used for the data obtained in this report.

EXPERIMENTAL PROCEDURE

There are two procedures for analyzing echo-reduction measurements with the modified Prony method. In the first, referred to as the two-window method, the projector is positioned 170 cm and the probe 15 cm in front of the panel. A USRD type F36 standard transducer is used as the projector while the probe is a USRD type H52 standard hydrophone. The projector is driven by a step sinusoidal signal of 1-ms duration that produces a 200-μs segment of incident signal, at the probe, followed by 200 μs of incident plus reflected signal before the arrival of the diffracted signal from the panel edges. This allows equal periods of the incident and reflected signals to be observed.

The waveform at the probe is sampled at 1 MHz, and approximately 50 to 100 waveforms are averaged to reduce the incoherent noise level. The waveform is then divided into two time windows--one containing only the incident signal, and the second containing the incident plus reflected portions of the signal. Figure (10) illustrates the waveform and the two time windows used in analyzing the waveform. Both time windows are analyzed by the modified Prony method to find the amplitudes of the steady-state driving frequency poles. A 15-pole expansion is used with three poles entered a priori. Two of the a priori poles are the complex conjugate pair representing the driving frequency, and the third a priori pole is a real pole associated with a high-pass RC filter on the input side.

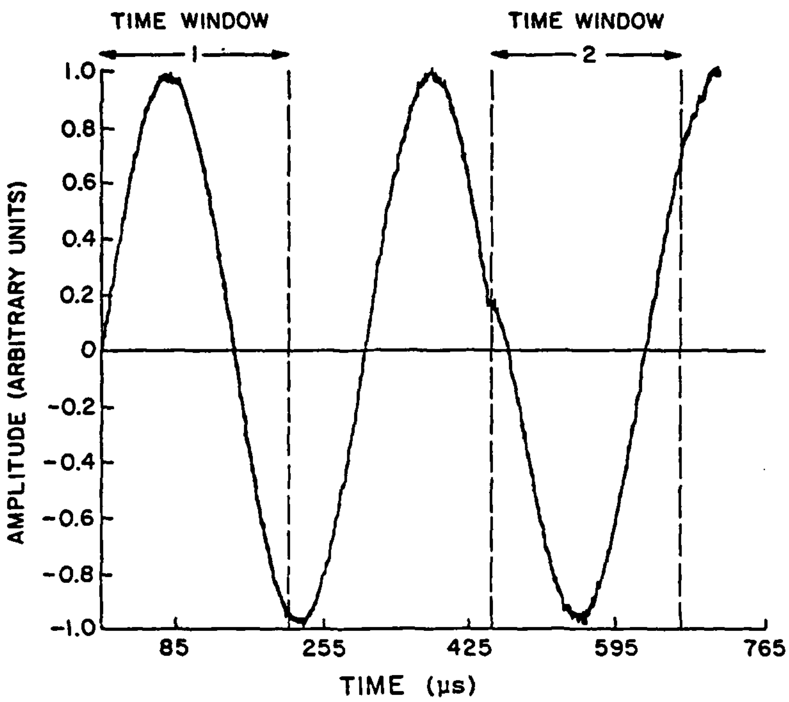

Fig. (10) - Illustration of two-window method. Two time windows are obtained from the waveform; the first containing only the incident signal and the second containing the incident plus reflected signal.

The complex amplitude of the incident portion is then phase shifted by an amount equal to the time separation of the two time windows and substracted from the complex amplitude of the incident plus reflected portion. This yields the complex amplitude of the steady-state driving frequency pole for the reflected signal. The echo reduction is then calculated as

$$\text{Echo reduction} = 20 \log {}^{A_I}/A_r \ , \tag{35}$$

where A_I and A_r are respectively the moduli of the amplitudes of the incident and reflected signals.

While the two-window method yields good results, it is not as accurate as the second method, to be discussed below, owing to phase errors. The algorithm does a much better job of finding the correct modulus of the amplitude than it does in finding the correct phase. This phase error introduces an error into the echo-reduction calculation when the incident amplitude is phase shifted and substracted from the incident plus reflected amplitude. Not all of the incident signal is cancelled, and the amplitude obtained does not represent the reflected signal only. The magnitude of the error will depend on the phase error and the relative phase of the direct and reflected signals. If the length of the time window is equivalent to at least one period of the driving frequency, frequencies above 5 kHz for a 200-μs time window, the phase error is negligible and the two-window method is sufficient. However, as the frequency is reduced, the phase error increases and an alternate method must be used.

The second method, referred to as the difference method, eliminates the effect of the phase error in the algorithm by directly subtracting out the incident signal. This method was basically described in connection with data acquisition for the section on the data increment. It consists of performing two separate measurements--one with and one without the acoustic panel in position. The two recorded waveforms are then directly subtracted to yield the reflected signal from the panel as illustrated in Fig. (11). Then the modified Prony method is used to analyze the incident and reflected waveforms separately to obtain the amplitudes of the driving frequency poles.

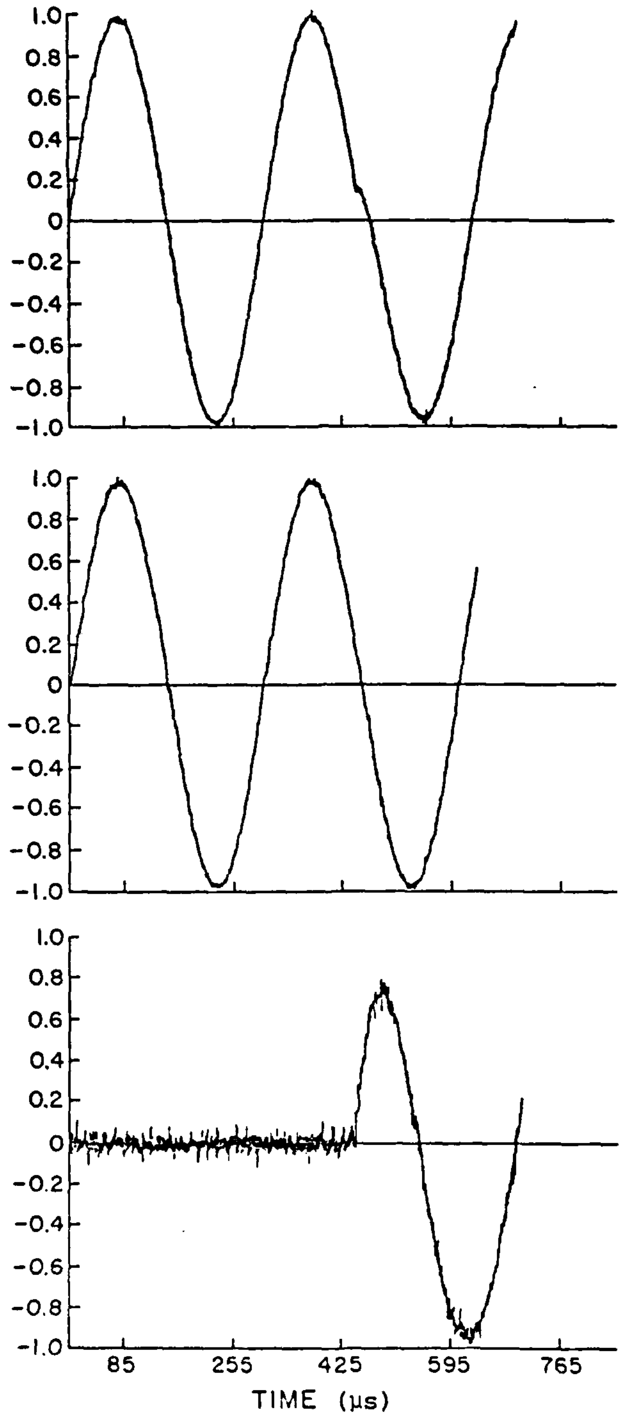

Fig. (11) - Sequence illustrating difference method. The top waveform was obtained with panel in position while the middle waveform was obtained without the panel. The lower waveform was obtained by direct substraction of the width and without panel waveforms.

The difference method has the additional advantage of allowing longer portions of the reflected signal to be observed. In the two-window method the optimum position of the probe is 15 cm in front of the panel since this allows equal segments of the incident and reflected signals to be observed. However, in the difference method the measurement performed without the panel produces the waveform for the incident signal. This allows the panel to be positioned close to the probe for the second measurement. With the probe 5 cm from the panel, 250 μs of reflected signal can be observed prior to the arrival of the diffracted signal from the panel edges. However, care must be taken to insure that the two measurements are identical. In addition a least-square subtraction should be used in obtaining the reflected signal to compensate for any gain and phase changes that may occur between measurements.

The disadvantages of the difference method are the additional time required for separate measurements and an inherent phase error due to digitizing the waveform. The time factor essentially doubles the time required to perform the measurements while the phase error becomes a problem only at high frequencies where the two-window method is accurate. This results in an obvious choice of using the two-window method, except at low frequencies (below 5 kHz) where the difference method is more accurate.

EXPERIMENTAL RESULTS

In Figs. (12), (13), and (14) the results of echo-reduction measurements of steel and aluminum panels have been plotted against theoretical curves. The measurements were performed on 0.95-cm-thick and 76.2-cm-square panels in the anechoic tank at USRD. The noise level in the anechoic tank during the measurements was approximately 40 dB below the incident signal level.

In Fig. (12) the measurements of the steel panel were processed by the difference method. Measurements were made with and without the panel in position, and 35 waveforms at each frequency were averaged. The no-panel waveforms were directly subtracted from the waveforms with the panel in position to obtain the waveform of the reflected signal. Each waveform was then processed with a 15-pole expansion that included the two driving frequency poles a priori. The results deviate from the theoretical curve by a few tenths of a dB.

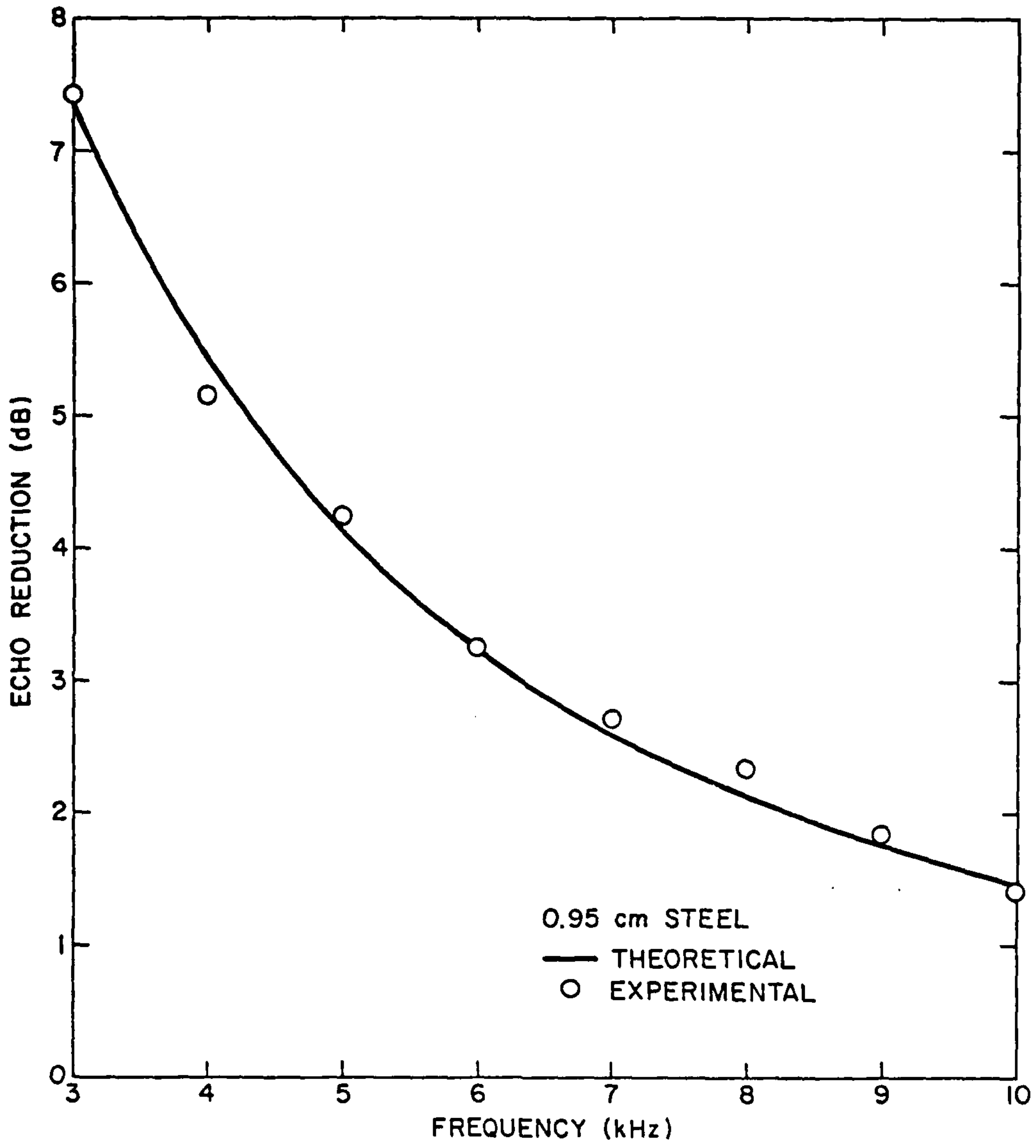

Fig. (12) - Results of Prony measurements of a 0.95-cm-thick steel panel processed by the difference method

In Figs. (13) and (14) the results of measurements on an aluminum panel have been plotted against theoretical curves. The aluminum plate was chosen since it has a larger echo reduction than the steel panel and was a better test of the method. The results in Fig. (13) were obtained in the same manner as the data for the steel panel except three a priori poles were used. The third a priori pole was associated with an RC filter on the input side of the electronics. The experimental results deviate from the theoretical curve by approximately 0.25 dB.

In Fig. (14) the measurements were processed by the two-window method. The probe was positioned 15 cm from the panel to provide equal segments of the incident and reflected signals and 50 separate measurements were averaged to obtain the waveforms. Each time window was processed with a 15-pole expansion that included three a priori poles. The results deviate from the theoretical curve by an average of 0.35 dB. However, the 3-kHz measurement deviates by 1.05 dB, a result explained by the previously described phase error associated with the two-window method.

In addition to the data presented here, measurements on a 1.27-cm-thick steel plate have been performed at 2 and 2.5 kHz with the difference method. These measurements deviated from the theoretical values by approximately 0.5 dB and indicate that the method is capable of performing accurate measurements down to 2 kHz.

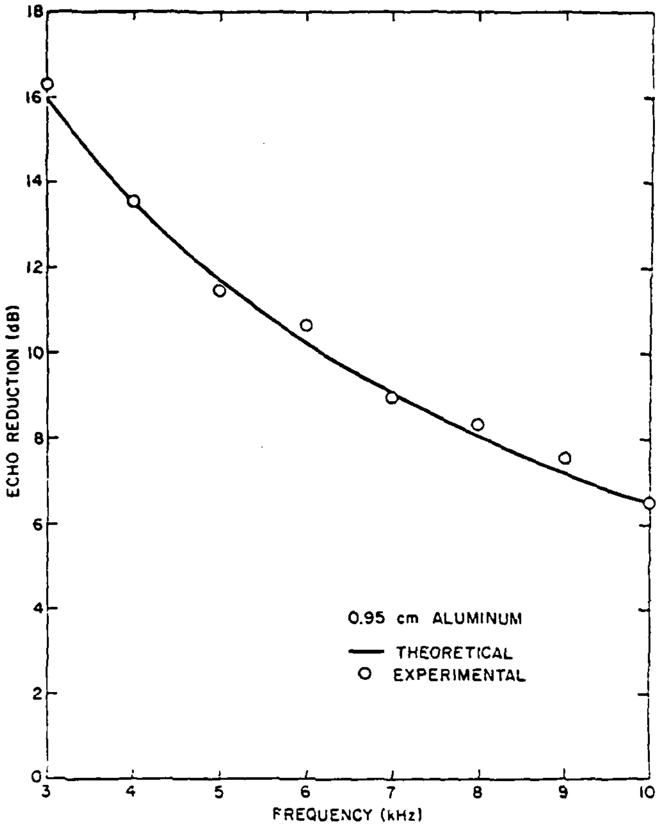

Fig. 13 - Results of Prony measurements of a 0.95-cm-thick aluminum panel processed by the difference method.

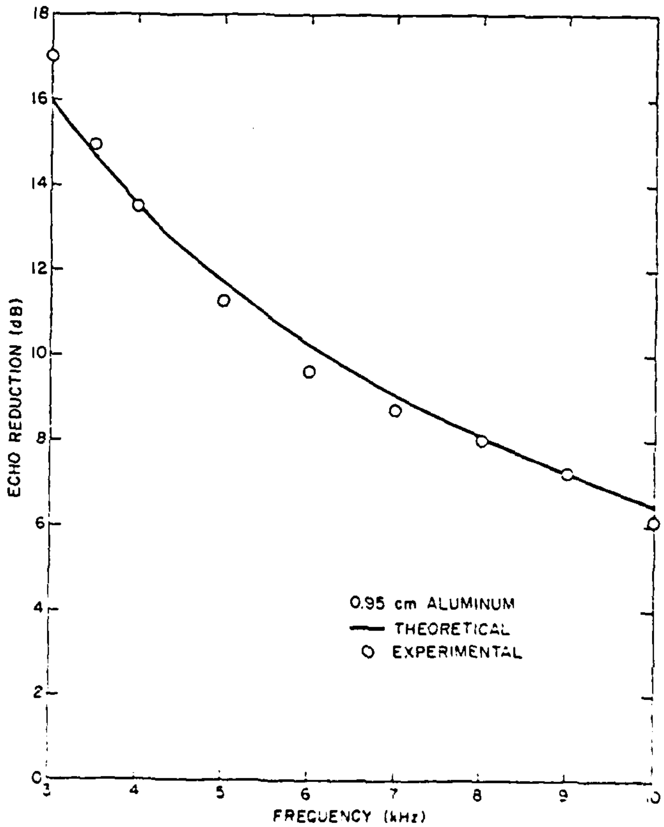

Fig. 14 - Results of Prony measurements of a 0.95-cm-thick aluminum panel processed by a two-window method.

CONCLUSION

The experimental results indicate that the modified Prony method is capable of making echo-reduction measurements down to 2 kHz on simple homogeneous panels with an error no greater than 0.5 dB. There have been no measurements, as yet, on high Q or lossy panels. However, these panels should not present an obstacle as long as the required number of terms in the expansion does not become too large.

ACKNOWLEDGMENTS

The author wishes to express his gratitude to the staff of the USRD's Anechoic Tank Facility for their help in obtaining the data for this report and to Dr. A. L. Van Buren of USRD's Methods Section for his patient guidance. Special acknowledgment is due to USRD's consultant Mr. A. Zed Robinson who is responsible for many ideas contained in this report and for reduction of the data.

REFERENCES

1 - R. Prony, "Essai expérimental et analytique sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alkool, á differentes températures," J.Ec. Polytech. 1, 24-76 (1795)

2 - F. B. Hildebrand, Advanced Calculus for Applications, 2nd Ed. (Prentice-Hall, New Jersey, 1963).

3 - M. L. Van Blaricum, "Techniques for extracting the complex resonances of a system directly from its transient reponse," Ph.D. dissertation, Electrical Engineering Det., Univ. of Illinois, Urbana, Dec. 1975.

4 - R. N. McDonough, "Representation and Analysis of Signals, Part XV. Matched Exponents for the Representation of Signals," John Hopkins U. Dep. Electr. Eng. Rep. (April 1963).

5 - L. G. Beatty, J. D. George and A. Zed Robinson, "Use of the complex exponential expansion as a signal representation for underwater acoustic calibration." J. Acoust. Soc. Am. 63(6), 1782-1794 (Jun 1978).

APPENDIX A

OVERVIEW

The following modified Prony program listing has been written in FORTRAN 4+ and is compatible with the Digital Electronics Corporation PDP 11/45 computer with the system RSX-11D. The program is designed to use data files, with a maximum of 1024 data points, that have the short IAG header format. Current dimension statements have limited the program to a maximum order of expansion of 15 and a maximum of 5 a priori poles.

APPENDIX B

LIST OF VARIABLES IN PRONY

J	-	number of data points in data file
ISP	-	initial start point in data file
NPTS	-	number of points in time window
IBD	-	data increment
NR	-	order of expansion
IA	-	number of a priori poles
DX	-	data file time increment
DEV	-	mean square deviation
ROOTS(I)	-	array of a priori poles (s plane)
ROOTZ(I)	-	array of z plane poles
DATA(I)	-	array of data points
COE(I)	-	coefficients of Prony polynomial
ACDEF(I)	-	array of amplitudes
COEFB(I)	-	coefficients of a priori polynomial
E(I,K)	-	matrix of Prony difference equations
F(I)	-	vector associated with Prony difference equations
A(I,K)	-	matrix of equations for amplitudes
R(I)	-	vector associated with A(I,J)

PRECEDING PAGE BLANK-NOT FILMED

APPENDIX C

INPUT EXAMPLE

```
RUN PRONY ⤸

ENTER FILE SPECS.
PRON.DHT;11 ⤸

ENTER INITIAL START POINT.252 ⤸

ENTER NUMBER OF POINTS IN TIME WINDOW.200 ⤸

ENTER BASIC DATA INCREMENT.(ODD INTEGER)  11 ⤸

ENTER ORDER OF EXPANSION.15 ⤸

DO YOU WISH TO ENTER APRIORI ROOTS? (Y/N) Y ⤸

ENTER # OF APRIORI POLES?2 ⤸

ENTER POLE VALUES AS

                REAL, IMAGINARY

 1    0.0,18849.5559 ⤸
 2    0.0,-18849.5559 ⤸

FINISHED
```

PRECEDING PAGE BLANK-NOT FILMED

Note: 1), All underlined portions are user supplied

2), ⤸ indicates 'RETURN'

APPENDIX D

OUTPUT EXAMPLE

DATE= 03-MAR-80

TIME= 10:38:05

INPUT DATA FILE INFORMATION

PRON.DHT;11

FILE HEADER INFORMATION

0 1024 1 0.00000 0.10000E-05 C 2

3KHZ

INITIAL START POINT= 252

TIME WINDOW= 0.20000E-03

BASIC DATA INCREMENT= 11

ORDER OF EXPANSION= 15

APRIORI POLES

#	REAL,	IMAG
1	0.00000E+00	0.18850E+05
2	0.00000E+00	-0.18850E+05

Z-PLANE POLES.			RESIDUES.		
1	0.99982E+00	0.18848E-01	1	0.60255E-01	-0.10084E+00
2	0.99982E+00	-0.18848E-01	2	0.60255E-01	0.10084E+00
3	0.99893E+00	0.62862E-01	3	-0.37841E-03	0.50109E-03
4	0.99893E+00	-0.62862E-01	4	-0.37840E-03	-0.50108E-03
5	0.10032E+01	0.10119E+00	5	-0.37830E-03	0.42420E-03
6	0.10032E+01	-0.10119E+00	6	-0.37830E-03	-0.42420E-03
7	0.97139E+00	0.14880E+00	7	-0.31909E-02	-0.21556E-03
8	0.97139E+00	-0.14880E+00	8	-0.31909E-02	0.21557E-03
9	0.98065E+00	0.18235E+00	9	-0.47842E-03	-0.60498E-03
10	0.98065E+00	-0.18235E+00	10	-0.47842E-03	0.60498E-03
11	0.96659E+00	0.25580E+00	11	0.16859E-03	0.74164E-04
12	0.96659E+00	-0.25580E+00	12	0.16859E-03	-0.74165E-04
13	0.95833E+00	0.20486E+00	13	0.15642E-02	-0.13310E-02
14	0.95833E+00	-0.20486E+00	14	0.15642E-02	0.13310E-02

MEAN SQUARE DEVIATION= 0.36180E-06

APPENDIX E

PRONY LISTING

```
C
C
C       PRONY MAIN PROGRAM
C
        COMPLEX ROOTS(10),ROOTZ(15),ACOEF(15),E(15,15),F(15)
        DIMENSION COEFB(16),A(15,15),R(15),DATA(1024),COE(16)
        BYTE TIM(8),DAT(9)
C
        CALL DATE(DAT)
        WRITE(2,500)DAT
        CALL TIME(TIM)
        WRITE(2,510)TIM
        CALL RFILES(DATA,J,DX)
C
C       OBTAIN INITIAL START POINT
C
1       WRITE(6,520)
        READ(5,530)ISP
        WRITE(2,535)ISP
C
C       CHECK THAT ISP IS GREATER THAN ZERO
C
        IF(ISP-1)5,10,10
5       WRITE(6,540)
        GOTO 1
C
C       OBTAIN NUMBER OF POINTS IN DATA WINDOW
C
10      WRITE(6,550)
        READ(5,560)NPTS
        TW=DX*NPTS
C
C       CHECK THAT WINDOW DOES NOT EXCEED DATA RANGE
C
        IF(ISP+NPTS-J-1)20,20,15
15      WRITE(6,570)
        GOTO 1
20      WRITE(2,580)TW
C
C       OBTAIN BASIC DATA INCREMENT
C
25      WRITE(6,590)
        READ(5,600)IBD
```

PRECEDING PAGE BLANK-NOT FILMED

```
C
C       CHECK THAT IBD IS AN ODD INTEGER
C
        IJ=IBD/2
        IF(2*IJ-IBD)35,30,30
30      WRITE(6,610)
        GOTO 25
C
C       OBTAIN ORDER OF EXPANSION
C
35      WRITE(6,620)
        READ(5,630)NR
        WRITE(2,640)NR
C
C       ARE THERE ANY APRIORI ROOTS?
C
        WRITE(6,650)
        READ(5,660)IZ
        IF(IZ-'Y')40,50,40
C
C       NO APRIORI ROOTS
C
40      IA=0
        COEFB(1)=1.0
        COEFB(2)=0.0
        COEFB(3)=0.0
C
C       CHECK THAT THERE ARE SUFFICIENT DATA POINTS
C
        IF(NPTS-NR*(IBD+1))45,70,70
45      WRITE(6,670)
        GO TO 10
C
C       CALL PRIORI FOR APRIORI ROOTS
C
50      CALL PRIORI(ROOTS,ROOTZ,COEFB,IA,DX,IBD)
C
C       CHECK THAT IA WAS NOT SET EQUAL TO ZERO IN PRIORI
C
        IF(IA)55,40,55
C
C       CHECK THAT THERE ARE SUFFICIENT DATA POINTS FOR THE
C       CASE WITH APRIORI ROOTS.
C
55      IF(NPTS-NR*(IBD+1)+IA)60,65,65
60      WRITE(6,670)
        GOTO 10
```

```
C
C       CHECK FOR CASE WHERE NUMBER OF APRIORI ROOTS=
C       ORDER OF EXPANSION
C
65      IF(NR-IA)66,110,70
66      NR=IA
        GOTO 110
70      CALL MATRIX(A,R,NR,IA,NPTS,COEFB,DATA,ISP,IBD,IL)
C
C       CHECK RETURN FROM MATRIX
C
        IF(IL-10)75,10,75
C
C       CHECK FOR THE CASE WHERE THERE IS ONLY ONE NON-APRIORI
C       ROOT
C
75      IF(NR-IA-1)65,80,85
C
C       OBTAIN SINGLE DESIRED ROOT
C
80      ROOTZ(IA+1)=CMPLX(-R(1)/A(1,1),0.0)
        GOTO 110
85      CALL SOLVER(A,R,COE,NR,IA,IB)
C
C       CHECK EXIT FROM SOLVER
C
        IF(IB)95,90,95
90      NR=NR-1
        WRITE(2,680)NR
        IF(NR-IA)65,110,70
95      CALL PRQD(COE,NR,IER,ROOTZ,IA)
C
C       CHECK RETURN FROM PRQD
C
        IF(IER)110,110,105
105     NR=NR-1
        GO TO 70
110     IF(IBD-1)25,120,115
C
C       CONVERT Z-PLANE ROOTS TO IBD=1
C
115     CALL ROOTC(ROOTZ,IBD,NR)
120     WRITE(2,700)
        DO 125 I=1,NR
        WRITE(2,710)I,ROOTZ(I)
125     CONTINUE
C
```

```
C     OBTAIN ACOEF
C
130   CALL RESIDU(DATA,E,F,ROOTZ,NR,NPTS,ISP)
      CALL SOLVE(E,F,ACOEF,NR,IB)
      IF(IB)140,135,140
135   CALL ROOTE(ROOTZ,NR)
      GOTO 130
140   WRITE(2,720)
      DO 150 I=1,NR
      WRITE(2,710)I,ACOEF(I)
150   CONTINUE
      CALL MSDEV(ISP,DATA,DEV,NPTS,ROOTZ,ACOEF,NR)
      WRITE(2,730)DEV
170   WRITE(6,750)
C
C     FORMAT STATEMENTS
C
500   FORMAT(/,16X,' DATE= ',9A1)
510   FORMAT(/,16X,' TIME= ',8A1)
520   FORMAT(/,'$ENTER INITIAL START POINT.')
530   FORMAT(I4)
535   FORMAT(/,10X,' INITIAL START POINT= ',I4)
540   FORMAT(/,' INITIAL START POINT MUST BE GREATER THAN ZERO.')
550   FORMAT(/,'$ENTER NUMBER OF POINTS IN TIME WINDOW.')
560   FORMAT(I4)
570   FORMAT(/,' DATA WINDOW EXCEEDS DATA FILE.')
580   FORMAT(/,16X,' TIME WINDOW= ',E12.5)
590   FORMAT(/,'$ENTER BASIC DATA INCREMENT.(ODD INTEGER)  ')
600   FORMAT(I3)
610   FORMAT(/,' BASIC DATA INCREMENT MUST BE ODD INTEGER.')
620   FORMAT(/,'$ENTER ORDER OF EXPANSION.')
630   FORMAT(I3)
640   FORMAT(/,16X,' ORDER OF EXPANSION= ',I3)
650   FORMAT(/,'$DO YOU WISH TO ENTER APRIORI ROOTS? (Y/N) ')
660   FORMAT(1A1)
670   FORMAT(/,' INSUFFICIENT NUMBER OFDATA POINTS.')
680   FORMAT(/,16X,' ORDER OF EXPANSION= ',I3)
700   FORMAT(/,10X,' Z-PLANE POLES.',/)
710   FORMAT(/,6X,I2,6X,E12.5,5X,E12.5)
720   FORMAT(/,10X,' RESIDUES.',/)
730   FORMAT(/,16X,' MEAN SQUARE DEVIATION= ',E12.5)
750   FORMAT(/,' FINISHED')
      CALL EXIT
      END
```

```
C
C   PRONY SUBROUTINE FOR ENTERING APRIORI POLES INTO VECTOR
C   OF S-PLANE POLES.
C   PROGRAM ALSO CONVERTS S-PLANE POLES TO Z-PLANE POLES
C   AND COMPUTES B COEFFICIENTS FOR PRONY DIFFERENCE EQUATIONS.
C   DIMENSIONING HAS LIMITED SUBROUTINE TO 5 POLES.
C
    SUBROUTINE PRIORI(ROOTS,ROOTZ,COEFB,IA,DX,IBD)
C
C   ROOTS=VECTOR OF S-PLANE POLES
C   ROOTZ=VECTOR OF Z-PLANE POLES
C   COEFB=VECTOR OF B COEFFICIENTS ORDERED FROM LOW TO HIGH
C   IA=NUMBER OF APRIORI ROOTS
C   DX=DATA FILE TIME INCREMENT
C   IBD=BASIC DATA INCREMENT
C
    COMPLEX ROOTS(5),ROOTZ(15),B(6),C1
    DIMENSION COEFB(6)
C
C   ENTER NUMBER OF ROOTS
C
    WRITE(6,100)
    READ(5,110)IA
    IF(IA)90,90,5
C
C   ENTER ROOTS AS COMPLEX NUMBERS
C
5   WRITE(6,120)
    WRITE(6,130)
    DO 10 I=1,IA
    WRITE(6,140)I
    READ(5,150)ROOTS(I)
10  CONTINUE
    WRITE(2,160)
    WRITE(2,170)
    DO 15 IX=1,IA
    WRITE(2,180)IX,ROOTS(IX)
15  CONTINUE
C
C   CONVERT TO Z-PLANE
C
    C=IBD*DX
    C1=CMPLX(C,0.0)
    DO 20 IX=1,IA
```

```
      ROOTZ(IX)=CEXP(ROOTS(IX)*C1)
20    CONTINUE
25    CONTINUE
C
C     COMPUTE B COEFFICIENTS,SET ALL TERMS IN B(I)= 1
C
      DO 30 I=1,6
      B(I)=CMPLX(1.0,0.0)
30    CONTINUE
C
C     CHECK VALUE OF IA AND SET INITIAL VALUES
C
      IF(IA-1)90,40,45
40    COEFB(1)=REAL(ROOTZ(1))
      COEFB(2)=1.0
      GO TO 90
45    B(1)=ROOTZ(1)*ROOTZ(2)
      B(2)=-(ROOTZ(1)+ROOTZ(2))
      IF(IA-2)90,65,50
C
C     ENTER LOOP FOR CALCULATING COEFFICIENTS
C
50    DO 60 K=3,IA
      DO 55 J=K,2,-1
55    B(J)=B(J-1)-ROOTZ(K)*B(J)
      B(1)=-B(1)*ROOTZ(K)
60    CONTINUE
65    DO 70 I=1,IA+1
      COEFB(I)=REAL(B(I))
70    CONTINUE
90    RETURN
C
C     FORMAT STATEMENTS
C
100   FORMAT(/,'$ENTER # OF APRIORI POLES?')
110   FORMAT(I2)
120   FORMAT(/,' ENTER POLE VALUES AS',/)
130   FORMAT(/,16X,' REAL, IMAGINARY',/)
140   FORMAT(/,'$'I2,3X)
150   FORMAT(2E12.5)
160   FORMAT(/,27X,' APRIORI POLES',/)
170   FORMAT(7X,' #',15X,' REAL,'6X,' IMAG',/)
180   FORMAT(/,7X,I2,13X,2E12.5)
      END
```

```
C
C         PRONY SUBROUTINE FOR GENERATING THE PRONY DIFFERENCE
C         EQUATIONS IN THE FORM A*(COE)=R, WHEN APRIORI POLES
C         (ROOTZ) ARE GIVEN.'A' IS GENERATED BY A VIRTUAL MATRIX
C         PREMULTIPLIED BY IT'S TRANSPOSE.
C
          SUBROUTINE MATRIX(A,R,NR,IA,NPTS,COEFB,DATA,ISP,IBD,IL)
C
C         A=MATRIX CONTAINING DIFFERENCE EQUATIONS
C         R=COLUMN VECTOR WHICH ARISES DUE TO THE CONSTRAINT
C           THAT THE HIGHEST BETA COEFFICIENT EQUAL ONE
C         NR=NUMBER OF POLES IN PRONY EXPANSION
C         IA=NUMBER OF APRIORI POLES
C         NPTS=NUMBER OF POINTS IN DATA
C         COEFB=COEFFICIENTS FROM APRIORI POLES
C         DATA=DATA FILE
C         ISP=INITIAL START POINT IN DATA
C         IBD=BASIC DATA INCREMENT
C         IL=RETURN CODE
C
          DIMENSION A(15,15),R(15),COEFB(6),DATA(1024)
C
C         DEFINE VARIABLE RANGE
C
          IC=NR-IA
          IR=NPTS-NR*IBD
          IS=ISP-1
C
C         CHECK FOR EXACTLY SOLVED CASE
C
          IN=IR-IC
          IF(IN)10,20,50
10        IL=10
          WRITE(6,500)
          RETURN
C
C         EXACTLY SOLVED CASE
C
20        DO 30 I=1,IC
          DO 30 J=I,IC
          A(I,J)=0.0
          DO 30 IK=1,IA+1
```

```
      A(I,J)=COEFB(IK)*DATA(I+IS+(IK-1)*IBD+(J-1)*IBD)+A(I,J)
   30 CONTINUE
      DO 40 I=1,IC
      R(I)=0.0
      DO 40 IK=1,IA+1
      R(I)=R(I)-COEFB(IK)*DATA(I+IS+(IK-1)*IBD+IC*IBD)
   40 CONTINUE
      GO TO 80
C
C     LEAST SQUARE TYPE SOLUTION
C
   50 DO 70 I=1,IC
      DO 70 J=I,IC
      A(I,J)=0.0
      R(I)=0.0
      DO 70 K=1,IR
      B1=0.0
      B2=0.0
      R1=0.0
      DO 60 IK=1,IA+1
      B1=COEFB(IK)*DATA(K+IS+(IK-1)*IBD+(J-1)*IBD)+B1
      B2=COEFB(IK)*DATA(K+IS+(IK-1)*IBD+(I-1)*IBD)+B2
      R1=COEFB(IK)*DATA(K+IS+(IK-1)*IBD+IC*IBD)+R1
   60 CONTINUE
      A(I,J)=B1*B2+A(I,J)
      R(I)=R(I)-R1*B2
   70 CONTINUE
   80 IF(IC-1)110,110,90
   90 DO 100 I=2,IC
      DO 100 J=1,I-1
      A(I,J)=A(J,I)
  100 CONTINUE
  110 IL=36
      RETURN
  500 FORMAT(/,' INSUFFICIENT NUMBER OF DATA POINTS,IN (MATRIX).')
      END
```

```
C
C
C        PRONY SUBROUTINE FOR SOLVING THE LEAST SQUARE EQUATIONS
C        GENERATED IN MATRIX TO FIND THE COEFFICIENTS OF THE
C        PRONY POLYNOMIAL
C
         SUBROUTINE SOLVER(A,R,COE,NR,IA,IB)
C
C
         DIMENSION A(15,15),R(15),COE(16),X(15),IKTA(15)
C
         IB=1
         N=NR-IA
         DO 10 I=1,N
         IKTA(I)=I
10       CONTINUE
         K=1
C
C        CHECK LEADING TERM
C
15       IF(A(K,K))30,20,30
20       CALL INTERD(A,R,IKTA,K,N,IC1)
         IF(IC1)30,25,30
25       IB=0
         RETURN
30       CONTINUE
C
C        DIVIDE ROWS BY LEADING TERM
C
         C1=A(K,K)
         R(K)=R(K)/C1
         DO 40 J=K,N
         A(K,J)=A(K,J)/C1
40       CONTINUE
C
C        SUBTRACT K ROW FROM ALL ROWS BELOW
C
         DO 50 I=K+1,N
         R(I)=R(I)-R(K)*A(I,K)
         C1=A(I,K)
         DO 50 J=K,N
         A(I,J)=A(I,J)-A(K,J)*C1
50       CONTINUE
```

```
      K=K+1
      IF(K-N)15,60,60
60    X(N)=R(N)/A(N,N)
      DO 70 I=N-1,1,-1
      X(I)=R(I)
      DO 70 J=N,I+1,-1
      X(I)=X(I)-X(J)*A(I,J)
70    CONTINUE
      DO 80 I=1,N
      J=IKTA(I)
      COE(J)=X(I)
80    CONTINUE
      COE(N+1)=1.0
      RETURN
      END
```

```
C
C
C       SUBROUTINE INTERCHANGES ROWS AND COLUMNS OF MATRIX A AND
C       VECTOR R WHILE KEEPING TRACK OF CHANGES IN VECTOR IKTA.
C
        SUBROUTINE INTERD(A,R,IKTA,K,N,IC1)
C
C
        DIMENSION A(15,15),R(15),X1T(15),X2T(15),IKTA(16)
C
C
        IC1=0
        IR1=K
        X1=0.0
        DO 10 I=K,N
        IF(ABS(X1)-ABS(A(K,I)))5,10,10
5       IC1=I
        X1=A(K,I)
10      CONTINUE
        IF(IC1)15,15,20
15      RETURN
20      I=IR1
        DO 30 J=1,N
        X1T(J)=A(I,J)
30      CONTINUE
        I=IC1
        DO 40 J=1,N
        X2T(J)=A(I,J)
40      CONTINUE
        I=IR1
        DO 50 J=1,N
        A(I,J)=X2T(J)
50      CONTINUE
        I=IC1
        DO 60 J=1,N
        A(I,J)=X1T(J)
60      CONTINUE
C
C       INTERCHANGE COLUMNS
C
        J=IR1
        DO 70 I=1,N
        X1T(I)=A(I,J)
```

```
 70     CONTINUE
        J=IC1
        DO 80 I=1,N
        X2T(I)=A(I,J)
 80     CONTINUE
        J=IR1
        DO 90 I=1,N
        A(I,J)=X2T(I)
 90     CONTINUE
        J=IC1
        DO 110 I=1,N
        A(I,J)=X1T(I)
 110    CONTINUE
        R1T=R(IR1)
        R2T=R(IC1)
        R(IR1)=R2T
        R(IC1)=R1T
        I=IKTA(IR1)
        J=IKTA(IC1)
        IKTA(IR1)=J
        IKTA(IC1)=I
        RETURN
        END
```

```
C
C
C        PRONY SUBROUTINE FOR CONVERTING THE ROOTZ FOUND WITH
C        THE BASIC DATA INCREMENT NOT EQUAL TO ONE (ROOTZ)**IBD
C        TO THE ROOTZ WITH BASIC DATA INCREMENT OF ONE.
C
         SUBROUTINE ROOTC(ROOTZ,IBD,NR)
C
C        ROOTZ=CONTAINS THE ROOTZ**IBD ON RETURN CONTAINS ROOTZ
C        IBD=BASIC DATA INCREMENT
C
         COMPLEX ROOTZ(15)
C
C
         DO 20 I=1,NR
         A=AIMAG(ROOTZ(I))
         B=REAL(ROOTZ(I))
         J=1.0E+3*A
         K=ININT(1.0E3*B)
         IF(J)10,5,10
5        IF(K)6,8,10
6        B=ABS(REAL(ROOTZ(I)))
         B=B**(1/FLOAT(IBD))
         ROOTZ(I)=CMPLX(-B,0.0)
8        GOTO 20
10       ROOTZ(I)=CEXP((CLOG(ROOTZ(I)))/IBD)
20       CONTINUE
         RETURN
         END
C
C        PRONY SUBROUTINE FOR DELETING ROOTS WHEN RESIDU FAILS
C
         SUBROUTINE ROOTE(ROOTZ,NR)
         COMPLEX ROOTZ(15)
C
C
         WRITE(6,100)
         DO 10 I=1,NR
         WRITE(6,110)I,ROOTZ(I)
10       CONTINUE
15       WRITE(6,120)
         READ(5,130)IX
```

```
      DO 20 I=IX,NR-1
      ROOTZ(I)=ROOTZ(I+1)
20    CONTINUE
      ROOTZ(NR)=CMPLX(0.0,0.0)
      NR=NR-1
      WRITE(6,140)
      READ(5,150)IJ
      IF(IJ-'Y')30,15,30
30    RETURN
C
C     FORMAT STATEMENTS
C
100   FORMAT(/,'1Z-PLANE POLES')
110   FORMAT(3X,I2,3X,E12.5,4X,E12.5)
120   FORMAT(/,'$WHICH POLE IS TO BE DELETED?')
130   FORMAT(I2)
140   FORMAT(/,'$DELETE ANOTHER?')
150   FORMAT(1A1)
      END
```

```
C
C       PRONY SUBROUTINE WHICH LOADS THE MATRIX E AND VECTOR F
C       WITH THE LEAST SQUARE EQUATIONS FOR CALCULATING THE
C       RESIDUES ASSOCIATED WITH THE POLES.
C
        SUBROUTINE RESIDU(DATA,E,F,ROOTZ,NR,NPTS,ISP)
C
C
        COMPLEX ROOTZ(15),A,B,E(15,15),F(15)
        DIMENSION DATA(1024)
C
C
        A=CMPLX(1.0,0.0)
        DO 10 I=1,NR
        DO 10 J=I,NR
        B=(CONJG(ROOTZ(I)))*ROOTZ(J)
        IF(AIMAG(B))5,1,5
1       IF(REAL(B)-1.0)5,2,5
2       E(I,J)=CMPLX(FLOAT(NPTS),0.0)
        GO TO 10
5       E(I,J)=(A-B**(NPTS))/(A-B)
10      CONTINUE
C
C
        DO 20 K=1,NR
        F(K)=CMPLX(DATA(ISP),0.0)
        DO 20 I=1,(NPTS-1)
        F(K)=DATA(ISP+I)*(CONJG(ROOTZ(K))**I)+F(K)
20      CONTINUE
        DO 30 I=2,NR
        DO 30 J=1,I-1
        E(I,J)=CONJG(E(J,I))
30      CONTINUE
        RETURN
        END
```

```
C
C        PRONY SUBROUTINE WHICH SOLVES THE LEAST SQUARE
C        EQUATIONS GENERATED IN RESIDU TO FIND THE RESIDUES.
C        SUBROUTINE SOLVES SIMULTANEOUS EQUATIONS WITH
C        COMPLEX COEFFICIENTS.
C
         SUBROUTINE SOLVE(E,F,ACOEF,NR,IB)
C
C
         COMPLEX E(15,15),F(15),ACOEF(15),X(15),C1,C2
         DIMENSION IKT(15)
C
C
         IB=1
C
C        FILL ARRAY TO KEEP TRACK OF ROW AND COLUMN
C        INTERCHANGES
C
         DO 10 I=1,NR
         IKT(I)=I
10       CONTINUE
         K=1
C
C        CHECK LEADING TERM
C
15       C3=REAL(E(K,K))
         IF(C3)34,20,34
C
C        INTERCHANGE ROWS AND COLUMNS
C
20       CALL INTERC(E,F,IKT,K,NR,IC1)
         IF(IC1)34,30,34
30       IB=0
         RETURN
C
C        DIVIDE ROWS BY LEADING TERM
C
34       N=K
35       C2=E(N,N)
         F(N)=F(N)/C2
         DO 40 J=K,NR
         E(N.J)=E(N,J)/C2
40       CONTINUE
```

```
      IF(NR-N)50,50,45
45    N=N+1
      GO TO 35
C
C     SUBTRACT K ROW FROM ALL ROWS BELOW
C
50    DO 60 I=K+1,NR
      F(I)=F(I)-F(K)*E(I,K)
      C2=E(I,K)
      DO 60 J=K,NR
      E(I,J)=E(I,J)-E(K,J)*C2
60    CONTINUE
      K=K+1
      IF(K-NR)15,70,70
70    X(NR)=F(NR)/E(NR,NR)
      DO 80 I=NR-1,1,-1
      X(I)=F(I)
      DO 80 J=NR,I+1,-1
      X(I)=X(I)-X(J)*E(I,J)
80    CONTINUE
      DO 90 I=1,NR
      J=IKT(I)
      ACOEF(J)=X(I)
90    CONTINUE
      RETURN
      END
```

```
C
C       SUBROUTINE INTERCHANGES ROWS AND COLUMNS OF MATRIX E
C       AND VECTOR F WHILE KEEPING TRACK OF CHANGES IN VECTOR
C       IKT.
C
        SUBROUTINE INTERC(E,F,IKT,K,NR,IC1)
C
C
        COMPLEX E(15,15),F(15),X1T(15),X2T(15),R1T,R2T
        DIMENSION IKT(15)
C
C
        IC1=0
        IR1=K
        X1=0.0
        DO 10 I=K,NR
        X2=REAL(E(K,I))
        IF(ABS(X1)-ABS(X2))5,10,10
5       IC1=I
        X1=X2
10      CONTINUE
        IF(IC1)15,15,20
15      RETURN
20      I=IR1
        DO 30 J=1,NR
        X1T(J)=E(I,J)
30      CONTINUE
        I=IC1
        DO 40 J=1,NR
        X2T(J)=E(I,J)
40      CONTINUE
        I=IR1
        DO 50 J=1,NR
        E(I,J)=X2T(J)
50      CONTINUE
        I=IC1
        DO 60 J=1,NR
        E(I,J)=X1T(J)
60      CONTINUE
C
C       INTERCHANGE COLUMNS
C
```

```
      J=IR1
      DO 70 I=1,NR
      X1T(I)=E(I,J)
70    CONTINUE
      J=IC1
      DO 80 I=1,NR
      X2T(I)=E(I,J)
80    CONTINUE
      J=IR1
      DO 90 I=1,NR
      E(I,J)=X2T(I)
90    CONTINUE
      J=IC1
      DO 110 I=1,NR
      E(I,J)=X1T(I)
110   CONTINUE
      R1T=F(IR1)
      R2T=F(IC1)
      F(IR1)=R2T
      F(IC1)=R1T
      I=IKT(IR1)
      J=IKT(IC1)
      IKT(IR1)=J
      IKT(IC1)=I
      RETURN
      END
```

```
C
C        PRONY SUBROUTINE FOR CALCULATING MEAN SQUARE DEVIATION
C        BETWEEN PRONY RECONSTRUCTED FILE AND ACTUAL DATA FILE.
C
         SUBROUTINE MSDEV(ISP,DATA,DEV,NPTS,ROOTZ,ACOEF,NR)
         DIMENSION DATA(1024)
         COMPLEX ROOTZ(15),ACOEF(15),B
C
C
         DEV=0.0
         DO 20 IX=ISP,NPTS+ISP-1
         IT=IX-ISP
         B=CMPLX(0.0,0.0)
         DO 10 IK=1,NR
         B=ACOEF(IK)*(ROOTZ(IK)**IT)+B
10       CONTINUE
         DEV=(REAL(B)-DATA(IX))**2+DEV
20       CONTINUE
         DEV=DEV/NPTS
         RETURN
         END
```

```
C   PRONY SUBROUTINE FOR READING INPUT DATA FILES IN
C   SINGLE PRECISION USING IAG HEADER FORMAT
C
C
    SUBROUTINE RFILES(DATA,J,DX)
C
C   J=NUMBER OF DATA POINTS IN FILE
C   DX=TIME INCREMENT
C
    DIMENSION DATA(1024),ICH(1),IHD(74)
    BYTE NAME(34),ITXT(148),CHAR(2)
    EQUIVALENCE(CHAR(1),ICH(1)),(IHD(1),ITXT(1))
C
C
    WRITE(6,100)
    READ(5,110)NA,NAME
    NAME(NA+1)=0
    WRITE(2,120)
    WRITE(2,130)(NAME(IX),IX=1,NA)
    OPEN(UNIT=4,NAME=NAME,TYPE='OLD',FORM='UNFORMATTED',READONLY
    IKC=1
    READ(4,END=180,ERR=190)I,J,K,SX,DX,ICH
    IKC=IKC+1
    WRITE(2,140)
    WRITE(2,150)I,J,K,SX,DX,CHAR(1),CHAR(2)
    NUM=CHAR(2)
    IF(NUM-1)15,15,5
5   DO 10 IX=1,NUM+1
    READ(4,END=180,ERR=190)IHD(IX)
10  CONTINUE
    WRITE(2,160)(ITXT(IX),IX=1,2*NUM)
15  IKC=IKC+1
    DO 20 IX=1,J
    READ(4,END=180,ERR=190)DATA(IX)
20  CONTINUE
    CALL CLOSE(4)
    RETURN
100 FORMAT(/,' ENTER FILE SPECS.')
110 FORMAT(Q,34A1)
120 FORMAT(/,16X,' INPUT DATA FILE INFORMATION',/)
130 FORMAT(16X,' ',<2*NA>A1)
140 FORMAT(/,16X,' FILE HEADER INFORMATION',/)
150 FORMAT(/,13X,3I6,F12.5,E12.5,4X,1A1,4X,I4,/)
```

```
160     FORMAT(16X,' '<2*NUM>A1)
180     WRITE(6,200)IKC
        GO TO 220
190     WRITE(6,210)IKC
200     FORMAT(' END OF FILE ON READ-PROG. EXIT',I3,/)
210     FORMAT(' ERR ON READ-PROG. EXIT',I3,/)
220     CALL EXIT
        END
```

```
      SUBROUTINE PROD(COE,NR,IER,ROOTZ,IA)  (see note at end of
C                                                this appendix)
C      DIMENSIONED DUMMY VARIABLES
      DIMENSION E(16),Q(16),COE(16),POL(16)
      COMPLEX ROOTZ(15)
C
C         NORMALIZATION OF GIVEN POLYNOMIAL
C            TEST OF DIMENSION
C         IR CONTAINS INDEX OF HIGHEST COEFFICIENT
      IER=0
      IC=NR-IA+1
      IR=IC
      EPS=1.E-6
      TOL=1.E-3
      LIMIT=10*IC
      KOUNT=0
    1 IF(IR-1)79,79,2
C
C         DROP TRAILING ZERO COEFFICIENTS
    2 IF(COE(IR))4,3,4
    3 IR=IR-1
      GOTO 1
C
C            REARRANGEMENT OF GIVEN POLYNOMIAL
C         EXTRACTION OF ZERO ROOTS
    4 O=1./COE(IR)
      IEND=IR-1
      ISTA=1
      NSAV=IR+1
      JBEG=1
C
C         Q(J)=1.
C         Q(J+I)=COE(IR-I)/COE(IR)
C         Q(IR)=COE(J)/COE(IR)
C         WHERE J IS THE INDEX OF THE LOWEST NONZERO COEFFICIENT
      DO 9 I=1,IR
      J=NSAV-I
      IF(COE(I))7,5,7
    5 GOTO(6,8),JBEG
    6 NSAV=NSAV+1
      Q(ISTA)=0.
      E(ISTA)=0.
      ISTA=ISTA+1
```

```
      GOTO 9
    7 JBEG=2
    8 Q(J)=COE(I)*0
      COE(I)=Q(J)
    9 CONTINUE
C
C              INITIALIZATION
      ESAV=0.
      Q(ISTA)=0.
   10 NSAV=IR
C
C          COMPUTATION OF DERIVATIVE
      EXPT=IR-ISTA
      E(ISTA)=EXPT
      DO 11 I=ISTA,IEND
      EXPT=EXPT-1.0
      POL(I+1)=EPS*ABS(Q(I+1))+EPS
   11 E(I+1)=Q(I+1)*EXPT
C
C          TEST OF REMAINING DIMENSION
      IF(ISTA-IEND)12,20,60
   12 JEND=IEND-1
C
C          COMPUTATION OF S-FRACTION
      DO 19 I=ISTA,JEND
      IF(I-ISTA)13,16,13
   13 IF(ABS(E(I))-POL(I+1))14,14,16
C
C          THE GIVEN POLYNOMIAL HAS MULTIPLE ROOTS, THE COEFFICIEN

C          THE COMMON FACTOR ARE STORED FROM Q(NSAV) UP TO Q(IR)
   14 NSAV=I
      DO 15 K=I,JEND
      IF(ABS(E(K))-POL(K+1))15,15,80
   15 CONTINUE
      GOTO 21
C
C              EUCLIDEAN ALGORITHM
   16 DO 19 K=I,IEND
      E(K+1)=E(K+1)/E(I)
      Q(K+1)=E(K+1)-Q(K+1)
      IF(K-I)18,17,18
C
C          TEST FOR SMALL DIVISOR
   17 IF(ABS(Q(I+1))-POL(I+1))80,80,19
   18 Q(K+1)=Q(K+1)/Q(I+1)
      POL(K+1)=POL(K+1)/ABS(Q(I+1))
```

```
      E(K)=Q(K+1)-E(K)
   19 CONTINUE
   20 Q(IR)=-Q(IR)
C
C              THE DISPLACEMENT EXPT IS SET TO 0 AUTOMATICALLY.
C              E(ISTA)=0.,Q(ISTA+1),...,E(NSAV-1),Q(NSAV),E(NSAV)=0

C              FORM A DIAGONAL OF THE QD-ARRAY.
C           INITIALIZATION OF BOUNDARY VALUES
   21 E(ISTA)=0.
      NRAN=NSAV-1
   22 E(NRAN+1)=0.
C
C              TEST FOR LINEAR OR CONSTANT FACTOR
C           NRAN-ISTA IS DEGREE-1
      IF(NRAN-ISTA)24,23,31
C
C           LINEAR FACTOR
   23 Q(ISTA+1)=Q(ISTA+1)+EXPT
      E(ISTA+1)=0.
C
C           TEST FOR UNFACTORED COMMON DIVISOR
   24 E(ISTA)=ESAV
      IF(IR-NSAV)60,60,25
C
C           INITIALIZE QD-ALGORITHM FOR COMMON DIVISOR
   25 ISTA=NSAV
      ESAV=E(ISTA)
      GOTO 10
C
C           COMPUTATION OF ROOT PAIR
   26 P=P+EXPT
C
C              TEST FOR REALITY
      IF(Q)27,28,28
C
C              COMPLEX ROOT PAIR
   27 Q(NRAN)=P
      Q(NRAN+1)=P
      E(NRAN)=T
      E(NRAN+1)=-T
      GOTO 29
C
C              REAL ROOT PAIR
   28 Q(NRAN)=P-T
      Q(NRAN+1)=P+T
      E(NRAN)=0.
```

```
C
C            REDUCTION OF DEGREE BY 2 (DEFLATION)
   29 NRAN=NRAN-2
      GOTO 22
C
C        COMPUTATION OF REAL ROOT
   30 Q(NRAN+1)=EXPT+P
C
C            REDUCTION OF DEGREE BY 1 (DEFLATION)
      NRAN=NRAN-1
      GOTO 22
C
C        START QD-ITERATION
   31 JBEG=ISTA+1
      JEND=NRAN-1
      TEPS=EPS
      TDELT=1.E-2
   32 KOUNT=KOUNT+1
      P=Q(NRAN+1)
      R=ABS(E(NRAN))
C
C            TEST FOR CONVERGENCE
      IF(R-TEPS)30,30,33
   33 S=ABS(E(JEND))
C
C        IS THERE A REAL ROOT NEXT
      IF(S-R)38,38,34
C
C        IS DISPLACEMENT SMALL ENOUGH
   34 IF(R-TDELT)36,35,35
   35 P=0.
   36 O=P
      DO 37 J=JBEG,NRAN
      Q(J)=Q(J)+E(J)-E(J-1)-O
C
C            TEST FOR SMALL DIVISOR
      IF(ABS(Q(J))-POL(J))81,81,37
   37 E(J)=Q(J+1)*E(J)/Q(J)
      Q(NRAN+1)=-E(NRAN)+Q(NRAN+1)-O
      GOTO 34
C
C        CALCULATE DISPLACEMENT FOR DOUBLE ROOTS
C            QUADRATIC EQUATION FOR DOUBLE ROOTS
C            X**2-(Q(NRAN)+Q(NRAN+1)+E(NRAN))*X+Q(NRAN)*Q(NRAN+1)

   38 P=0.5*(Q(NRAN)+E(NRAN)+Q(NRAN+1))
      O=P*P-Q(NRAN)*Q(NRAN+1)
```

```
      T=SQRT(ABS(O))
C
C         TEST FOR CONVERGENCE
      IF(S-TEPS)26,26,39
C
C         ARE THERE COMPLEX ROOTS
   39 IF(O)43,40,40
   40 IF(P)42,41,41
   41 T=-T
   42 P=P+T
      R=S
      GOTO 34
C
C         MODIFICATION FOR COMPLEX ROOTS
C         IS DISPLACEMENT SMALL ENOUGH
   43 IF(S-TDELT)44,35,35
C
C         INITIALIZATION
   44 O=Q(JBEG)+E(JBEG)-P
C
C         TEST FOR SMALL DIVISOR
      IF(ABS(O)-POL(JBEG))81,81,45
   45 T=(T/O)**2
      U=E(JBEG)*Q(JBEG+1)/(O*(1.+T))
      V=O+U
      KOUNT=KOUNT+2
C
C         THREEFOLD LOOP FOR COMPLEX DISPLACEMENT
      DO 53 J=JBEG,NRAN
      O=Q(J+1)+E(J+1)-U-P
C
C         TEST FOR SMALL DIVISOR
      IF(ABS(V)-POL(J))46,46,49
   46 IF(J-NRAN)81,47,81
   47 EXPT=EXPT+P
      IF(ABS(E(JEND))-TOL)48,48,81
   48 P=0.5*(V+O-E(JEND))
      O=P*P-(V-U)*(O-U*T-O*W*(1.+T)/Q(JEND))
      T=SQRT(ABS(O))
      GOTO 26
C
C             TEST FOR SMALL DIVISOR
   49 IF(ABS(O)-POL(J+1))46,46,50
   50 W=V*O/V
      T=T*(V/O)**2
      Q(J)=V+W-E(J-1)
      U=O.
```

```
      IF(J-NRAN)51,52,52
   51 U=Q(J+2)*E(J+1)/(O*(1.+T))
   52 V=O+U-W
C
C        TEST FOR SMALL DIVISOR
      IF(ABS(Q(J))-POL(J))81,81,53
   53 E(J)=W*V*(1.+T)/Q(J)
      Q(NRAN+1)=V-E(NRAN)
   54 EXPT=EXPT+P
      TEPS=TEPS*1.1
      TDELT=TDELT*1.1
      IF(KOUNT-LIMIT)32,55,55
C
C        NO CONVERGENCE WITH FEASIBLE TOLERANCE
C           ERROR RETURN IN CASE OF UNSATISFACTORY CONVERGENCE
   55 IER=1
C
C        REARRANGE CALCULATED ROOTS
   56 IEND=NSAV-NRAN-1
      E(ISTA)=ESAV
      IF(IEND)59,59,57
   57 DO 58 I=1,IEND
      J=ISTA+I
      K=NRAN+1+I
      E(J)=E(K)
   58 Q(J)=Q(K)
   59 IR=ISTA+IEND
C
C        NORMAL RETURN
   60 IR=IR-1
      IF(IR)78,78,61
C
C        REARRANGE CALCULATED ROOTS
   61 DO 62 I=1,IR
      Q(I)=Q(I+1)
   62 E(I)=E(I+1)
C
C        CALCULATE COEFFICIENT VECTOR FROM ROOTS
      POL(IR+1)=1.
      IEND=IR-1
      JBEG=1
      DO 69 J=1,IR
      ISTA=IR+1-J
      O=0.
      P=Q(ISTA)
      T=E(ISTA)
      IF(T)65,63,65
```

```
C
C         MULTIPLY WITH LINEAR FACTOR
   63 DO 64 I=ISTA,IR
      POL(I)=Q-P*POL(I+1)
   64 Q=POL(I+1)
      GOTO 69
   65 GOTO(66,67),JBEG
   66 JBEG=2
      POL(ISTA)=0.
      GOTO 69
C
C         MULTIPLY WITH QUADRATIC FACTOR
   67 JBEG=1
      U=P*P+T*T
      P=P+P
      DO 68 I=ISTA,IEND
      POL(I)=Q-P*POL(I+1)+U*POL(I+2)
   68 Q=POL(I+1)
      POL(IR)=Q-P
   69 CONTINUE
      IF(IER)78,70,78
C
C         COMPARISON OF COEFFICIENT VECTORS, IE. TEST OF ACCURACY
   70 P=0.
      DO 75 I=1,IR
      IF(COE(I))72,71,72
   71 Q=ABS(POL(I))
      GOTO 73
   72 Q=ABS((POL(I)-COE(I))/COE(I))
   73 IF(P-Q)74,75,75
   74 P=Q
   75 CONTINUE
      IF(P-TOL)77,76,76
   76 IER=-1
   77 Q(IR+1)=P
      E(IR+1)=0.
   78 DO 100 I=IA+1,NR
      ROOTZ(I)=CMPLX(Q(I-IA),E(I-IA))
  100 CONTINUE
      RETURN
C
C         ERROR RETURNS
C             ERROR RETURN FOR POLYNOMIALS OF DEGREE LESS THAN 1
   79 IER=2
      IR=0
      RETURN
```

```
C              ERROR RETURN IF THERE EXISTS NO S-FRACTION
   80 IER=4
      IR=ISTA
      GOTO 60
C
C              ERROR RETURN IN CASE OF INSTABLE QD-ALGORITHM
   81 IER=3
      GOTO 56
      END
```

<u>Re info on page 55</u>:

Taken from page 183
IBM Application Program
System/360 Scientific Subroutine Package
Version III
Programmer's Manual
Program Number 360A-CM-03X
Fifth Ed (1970)
IBM Corp., Technical Publications Dept.
White Plains, NY 10601